



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/997,801	11/30/2001	Eric Ledoux	MICR0218	4983
27792	7590	01/07/2005	EXAMINER	
MICROSOFT CORPORATION			PHAN, TAM T	
LAW OFFICES OF RONALD M. ANDERSON				
600 108TH AVENUE N.E., SUITE 507			ART UNIT	PAPER NUMBER
BELLEVUE, WA 98004			2144	

DATE MAILED: 01/07/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	09/997,801	LEDOUX ET AL.	
	Examiner Tam (Jenny) Phan	<b>Art Unit</b> 2144	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 04 October 2002.

2a) This action is FINAL.                    2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-34 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 1-34 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 30 November 2001 is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All    b) Some \* c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____

### **DETAILED ACTION**

1. This application has been examined. Pre-amendment A received on 10/04/2004 has been entered. Claims 1-34 are presented for examination.

#### ***Priority***

2. No priority claims have been made.
3. The effective filing date for the subject matter defined in the pending claims in this application is 11/30/2001.

#### ***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-34 are rejected under 35 U.S.C. 102(b) as being anticipated by Brown et al. (U.S. Patent Number 6,067,551), hereinafter referred to as Brown.
6. Regarding claim 1, Brown disclosed a method for maintaining synchronization of data stored on a server, where components of the data are modifiable on clients that are coupled to the server over a network and wherein modification to the components of the data on the clients can be uploaded to the server, comprising the steps of: (a) associating a version identifier [MCVI] with the data, said version identifier being incremented each time that a change to any component of the data occurs on the server (Figure 3 sign 70, Figure 4 sign 835, column 16 lines 5-16); (b) each time that a component of the data is modified on the server, assigning to the component the value of the version identifier that was current at the time the component was modified on the server (column 12 lines 9-24, column 16 lines 5-16); and (c) detecting a

proactive collision between a component of the data just downloaded to any client and a modified version of said component that was previously downloaded and modified by a user on said client, as a function of the values of version identifiers associated with the component downloaded and the modified version of the component, causing an indication of the proactive collision to be provided to the user, enabling the user to resolve the proactive collision (Figure 2G, Figure 6, column 13 lines 50-56, column 13 line 66-column 14 line 19, column 14 line 53-column 15 line 23).

7. Regarding claim 2, Brown disclosed a method further comprising the step of detecting reactive collisions between corresponding components of the data that are concurrently uploaded to the server from a plurality of clients if uploading of a corresponding component by one client is completed before that by another client, detection of a reactive collision causing the step (c) to be repeated for the other client (Figure 4 sign 840, Figure 2G, column 16 lines 5-16).

8. Regarding claim 3, Brown disclosed a method wherein the step of detecting a proactive collision comprises the step of automatically determining if the value of the version identifier of the component of the data just downloaded is different than the value of the version identifier of the modified component (column 12 lines 40-51, column 13 line 51-56).

9. Regarding claim 4, Brown disclosed a method wherein if there is an indication of that a proactive collision has occurred, the step of enabling the user to resolve the proactive collision comprises one of the steps of: (a) overwriting the modified version of the component with the component that was just downloaded (column 13 line 66-column 14 lines 29); and (b) uploading the modified version of the component to the server, so that a corresponding component on the server that was changed since the previous version of the component was downloaded and

subsequently modified by the user, is overwritten with the modified version (column 12 lines 52-67, column 14 lines 19-29).

10. Regarding claim 5, Brown disclosed a method further comprising the steps of: (a) enabling a new component of the data to be created on the client; and (b) enabling the new component to be uploaded to the server (column 15 line 49-16).

11. Regarding claim 6, Brown disclosed a method wherein each time that a client connects in communication with the server, further comprising the steps of: (a) downloading from the server to the client, each component for which the version identifier of said component on the server is greater than that of a corresponding component on the client (column 12 lines 9-24); (b) downloading an identification of each component of the data on the server, if a component has been deleted from the data on the server after the client was last synchronized with the server (column 18 lines 64-67); automatically overwriting each component on the client that has not been modified with a corresponding component that was downloaded from the server, if the version identifier for the component that was just downloaded is greater than that of the component already on the client (column 14 lines 19-29, column 15 lines 2-23); and (d) automatically deleting each component on the client that was deleted on the server since the client was last synchronized with the server (column 15 lines 9-23).

12. Regarding claim 7, Brown disclosed a method further comprising the step of maintaining on each client: (a) a server cache in which components most recently downloaded from the server are stored, and (b) a client store in which components of the data that have been modified on the client, but not yet uploaded to the server are stored (Figure 3, column 11 lines 43-57).

13. Regarding claim 8, Brown disclosed a method further comprising the step of maintaining on the server a unique identification for each object comprising the data stored on the server (Figure 3 sign 40).

14. Regarding claim 9, Brown disclosed a method wherein each time that a reactive collision is detected, causing step (c) to be repeated for the other client results in a proactive collision being detected between the component on the server just uploaded by said one client and the corresponding component that was being uploaded by the other client (Figure 2G, Figure 6, column 3 lines 48-59, column 13 lines 66-column 14 line 18).

15. Regarding claim 10, the memory medium having machine instructions that are readable by a computing device for maintaining synchronization data corresponds to the method for maintaining synchronization data of claim 1, and thus is rejected using the same rationale.

16. Regarding claim 11, Brown disclosed a method for maintaining synchronization of data stored on a server, said data being accessible by a plurality of clients at times coupled in communication with the server and able to download the data to be modified and to upload changes to the data to the server, said data including a plurality of nodes, comprising the steps of:

(a) assigning to the data a version identifier that is incremented each time any node of the data is modified on the server (Figure 2C sign 295, Figure 4 sign 835); (b) associating a value of the version identifier with each node, said value that is thereby associated corresponding to that of the version identifier then assigned to the data when the node was last modified on the server (Figure 2C, Figure 2F, Figure 3C signs 72, 74, column 5 lines 40-58); (c) enabling nodes that have been modified on the server since said nodes were previously downloaded by any client, to be downloaded to said client (column 3 lines 48-65, column 4 lines 14-27, (d) enabling nodes that were downloaded from the server by any client to be modified on said client, producing

Art Unit: 2144

modified nodes (column 4 lines 14-27, column 12 lines 17-24); (e) enabling the modified nodes to be uploaded from each client to the server (column 12 lines 52-67, column 14 lines 19-29); (f) detecting and providing an indication on each client of any proactive collision between a node that has just been downloaded from the server to the client and a corresponding node that was previously downloaded by the client and has been modified on the client, the proactive collision being detected as a function of the version identifiers associated with the node that has just been downloaded and the node that has been modified on the client (column 12 lines 40-51, column 13 lines 50-65); (g) detecting any reactive collision between corresponding modified nodes that were separately modified on two or more clients and which are being uploaded by the two or more clients, as a function of the version identifiers associated with the nodes that are being uploaded (column 13 lines 50-65); and (h) if a reactive collision is detected is step (g), repeating steps (e) - (h) (Figure 2G).

17. Regarding claim 12, Brown disclosed a method wherein the step of detecting the reactive collision occurs when the server detects that the version identifier of a node being uploaded by a client is different than a corresponding node now on the server, indicating that another client completed uploading of the corresponding node now on the server while said client was uploading said node (column 12 lines 40-51, column 13 line 51-56).

18. Regarding claim 13, Brown disclosed a method wherein before each download of nodes from the server to the clients occurs, further comprising the steps on each client, of: (a) conveying the version identifier for a class of nodes on the client to the server to indicate a version of the nodes in the class that were last downloaded from the server to the client (Figure 3, column 12 lines 9-16); (b) sending any nodes on the server to client, for which the version identifier associated therewith indicates the node on the server is a later version than the version

identifier of the class on the client (column 12 lines 9-24, column 18 lines 64-67); and (c) providing an indication of nodes remaining on the server if any node has been deleted on the server after the client was last synchronized with the server (column 14 line 55-column 15 line 23).

19. Regarding claim 14, Brown disclosed a method further comprising the step of: automatically overwriting each node not yet modified on the client with a corresponding node downloaded from the server; and deleting each node on the client that was indicated as having been deleted on the server (column 15 lines 9-23, column 16 lines 5-16).

20. Regarding claim 15, Brown disclosed a method further comprising the step of maintaining on each client: (a) a cache in which are stored a latest version of nodes most recently downloaded from the server; and (b) a storage containing all nodes modified on the client, but not yet uploaded to the server (Figure 3, column 11 lines 43-57).

21. Regarding claim 16, Brown disclosed a method wherein if there is an indication of a proactive collision being detected on a client, further comprising the step of enabling a user to elect one of the steps of: (a) overwriting the modified node on the client with the corresponding node just downloaded from the server; and (b) upload the modified node to the server, overwriting the corresponding node on the server (column 14 line 55-column 15 line 23).

22. Regarding claim 17, Brown disclosed a method further comprising the step of enabling new nodes to be uploaded from any of the clients to the server (column 16 lines 5-16).

23. Regarding claim 18, the memory medium having machine instructions that are readable by a computing device for maintaining synchronization data corresponds to the method for maintaining synchronization data of claim 11, and thus is rejected using the same rationale.

24. Regarding claim 19, Brown disclosed a method for maintaining synchronization of data transferred between a storage computing device and a plurality of remote computing devices that are at times coupled in data communication with the storage computing device to enable modification of the data, said data including a plurality of nodes, the method comprising the steps of: (a) assigning an identifier to the data (Figure 3, column 11 lines 42-57, column 12 lines 9-24); (b) changing the identifier each time that any node of the data is modified on the storage computing device so that the identifier indicates a version of the data that are recently stored on the storage computing device at that time (Figure 2C sign 295, column 12 lines 52-67); (c) associating a value of the identifier with each node stored on the storage computing device, said value indicating the version of the data at the time when the node was last modified on the storage computing device (Figure 3, column 11 lines 42-57, column 12 lines 9-24); (d) downloading a current identifier for the data and the plurality of nodes to any of the plurality of remote computing device that has requested transfer of the data from the storage computing device, for modification on the remote computing device, said current identifier being retained in association with the nodes that are downloaded to indicate a version of the nodes that were thus downloaded; enabling the nodes downloaded to be modified on any remote computing device having rights to do so (Figure 3, column 11 lines 42-57, column 12 lines 9-24); (f) at each subsequent time that one of the plurality of remote computing devices to which the nodes were downloaded in step (e) is coupled in data communication with the storage computing device for synchronization the date transferring the version indicator associated with the data that are retained on said one of the plurality of remote computing devices to the storage computing device (column 17 lines 1-13); (g) while synchronizing the data, downloading from the storage computing device to said one of the plurality of remote computing devices, each node of the data

for which the identifier associated with the node on the storage computing devices indicates that said node is a later version than indicated by the identifier associated with data previously downloaded from the storage computing device to said one of the remote computing devices, thereby updating the nodes on said one of the plurality of remote computing devices, but retaining any modified nodes (column 12 lines 9-24, lines 52-67); (h) detecting whether a node just downloaded in step (g) was modified on the storage computing device since a time that said node was previously downloaded and then modified to produce a modified node on said one of the plurality of remote computing devices, by comparison of the identifiers associated with the corresponding nodes, and if so, providing an indication thereof to a user of said one of the plurality of remote computing devices (column 12 lines 40-67, column 13 line 57-column 14 line 18); (i) enabling modified nodes to be uploaded to the storage computing device, along with the identifiers associated with the modified nodes (column 12 lines 40-67); and (j) detecting whether a newer modified node has been uploaded to the storage computing device before uploading of a modified node in step (i) is completed, and if so, repeating steps (h) - (i) (Figure 2G).

25. Regarding claim 20, Brown disclosed a method further comprising the step of enabling the user of said one of the plurality of remote computing devices to respond to said indication by electing one of the steps of: (a) overwriting the modified node on said one of the remote computing devices with the node just downloaded; and (b) uploading the modified version to the storage computing device, thereby overwriting the corresponding node on the storage computing device with the modified node and causing a change in the identifier associated with the data on the storage computing device (column 14 line 55-column 15 line 23)..

26. Regarding claim 21, Brown disclosed a method wherein during synchronizing, further comprising the step of downloading from the storage computing device to said one of the remote

computing devices a list identifying all nodes on the storage computing device, if a node has been deleted on the storage computing device after any nodes were downloaded to said one of the remote computing devices from the server computing device, causing nodes that were deleted on the server computing device to also be deleted on said one of the remote computing devices (column 15 lines 9-23, column 16 lines 5-16).

27. Regarding claim 22, Brown disclosed a method wherein on each of the plurality of remote computing devices, further comprising the steps of: (a) maintaining a cache for storing the nodes just downloaded from the storage computing device and the identifier associated with the nodes; and (b) maintaining a storage for each node that is modified on the remote computing device (Figure 3, column 11 lines 43-57, column 12 lines 16-24, column 14 lines 19-29).

28. Regarding claim 23, Brown disclosed a method wherein during synchronization, any node that has not been modified on said one of the remote computing devices since a previous synchronization is automatically overwritten with a corresponding node downloaded from the storage computing device (column 14 line 55-column 15 line 23).

29. Regarding claim 24, the memory medium having machine instructions that are readable by a computing device for maintaining synchronization data corresponds to the method for maintaining synchronization data of claim 19, and thus is rejected using the same rationale.

30. Regarding claims 25-30, the system for maintaining synchronization data corresponds to the method for maintaining synchronization data of claims 11-17, and thus these claims are rejected using the same rationale.

31. Regarding claim 31, Brown disclosed a method further comprising the step of enabling modification to be made by a user to the components of the data on a client while the client is not coupled to the server over the network, the modifications being subsequently uploaded to the

server while the client is then coupled to the server over the network (column 4 lines 14-27, column 12 lines 17-24, column 13 lines 30-47).

32. Regarding claim 32, Brown disclosed a method wherein the step of enabling the nodes that were downloaded from the server by any client to be modified on said client includes the step of enabling the nodes to be modified while the client is not connected to the server (column 4 lines 14-27, column 12 lines 17-24, column 13 lines 30-47).

33. Regarding claim 33, Brown disclosed a method wherein the step of enabling the nodes that are downloaded from the storage device to be modified by any remote computing device having the rights to do so includes the step of enabling the nodes to be modified while the remote computing device is not connected to the storage device (column 4 lines 14-27, column 12 lines 17-24, column 13 lines 30-47).

34. Regarding claim 34, the system for maintaining synchronization data corresponds to the method for maintaining synchronization data of claim 32, and thus is rejected using the same rationale.

35. Since all the limitations of the claimed invention were disclosed by Brown, claims 1-34 are rejected.

### ***Conclusion***

36. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Zollinger et al. (U.S. Patent Number 6,321,236) disclosed a method, computer program product, and system that allows changes made to an original database table found on a server computer to be reflected in client copies of the database table based on intermittent client requests for synchronization. A server makes periodic updates of table

differences between a current table receiving database change events and a reference table. Each client copy of a database table and update created by the server has a sequential version number associated therewith. The server will compare the version number of a client copy of a database table with the most recent version number of the table on the server to determine which updates need be applied in order to make the client copy current. The instructions are transmitted to the client (along with the new version number) so that the client may operate the database engine to apply the instructions for making the database table current with the original managed on the server. A client will initially receive a client copy of a database table having a particular version identifier, such as a version number, date stamp, etc. At some later time, the client will reconnect with the server to request synchronization of the client copy of the database table to make it current with the original database table that is on the server. The version identifier of the client copy of the database engine is accessed and all intervening updates are then translated into instructions that are understood by the type of database engine run on the client system. This allows the client copy of the database table to be made current with the original database table found on the server by the particular database engine running on the client system. For a sequentially numbered version number used as a version identifier, all updates having a larger number than that of the client copy of the database table are used to make the client copy current. The client copy of the database table is then given the latest version identifier and is considered current. Depending on when or how often a client connects with the server, one or multiple updates may be used in order to make the client copy of the database table current. The version identifier may be other than the major/minor revision version number explained above. Another version

identifier could be a date or time stamp that may be directly compared with other date or time stamps to determine which updates are needed to make a database table current. Furthermore, the date or time stamp may be combined with other version information such as the major/minor revision version numbering explained previously. For example, a client could simply ask for all updates since a certain date without tracking which version number it actually has. The server could compare the date to a file creation dates for the updates (if stored in a file) or other date and time stamp information in order to assure the correct updates are used to make the client copy of the database table current.

b. Arun et al. (U.S. Patent Number 6,631,386) disclosed a version control system is described for use in connection with a database management system to facilitate versioning of a database table, the system including a database table and a version control module. The database table comprises a plurality of records, each record including at least one data field for storing user data and at least some of the records including a version control field including version control information. The version control module is configured to, in response to a user query related to the database table and related to a version, generate an augmented query for processing by the data base management system, the augmented query relating to the user query and the version control information. The version control module facilitates association of versions of the database with respective ones of a hierarchy of states and allows conflicts there between to be resolved, data to be posted from child states to respective parent states in the hierarchy, and referential constraints between tables to be preserved.

c. Multer et al. (U.S. Patent Number 6694336) disclosed a method includes the steps of providing a data store associated with the first system and including information

representing data in the data files at a previous time state; generating first difference information upon a change to the data files by comparing the change to the data store; receiving second difference information for a subset of said data files from a second system; and applying said difference information to said subset of said data files. In many instances, the data to be synchronized is generally in the form of text data such as records of addresses, contact information, calendar information, notes and other types of contact information. In certain instances, data to be synchronized will be binary format of executable files or word processor-specific documents. In many cases where document synchronization is required, the synchronization routine simply determines whether or not the documents in question have changed, and uses a time-based representation to determine which of the two files is newer, and replaces the older file with the newer file to achieve synchronization, as long as the older of the two files was in fact not changed. This is the model used in the familiar "Briefcase" function in Microsoft Windows-based systems. If both files have changed, then the synchronization routine presents the option of conflict resolution to the user.

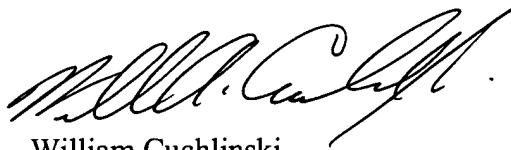
37. Refer to the enclosed PTO-892 for details and complete listing of other pertinent prior art of record.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tam (Jenny) Phan whose telephone number is (571) 272-3930. The examiner can normally be reached on M-F 9:00-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Cuchlinski can be reached on (571) 272-3925. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



William Cuchlinski  
SPE  
Art Unit 2144  
(571) 272-3925

tp  
December 30, 2004